This homework is due Monday 4/4 @ 10:30 PM.
Please submit your solutions as a PDF (written in \LaTeX) to PWeb.

This homework is broken up into two parts:

1. A pair of required problems covering mapping reducibility.

2. Some optional problems reinforcing our discussion of Turing machines and undecidability as well as reviewing difficult concepts from the first exam.

The required problems are just that—required for this homework. The optional problems are extra credit problems to help you prepare for the upcoming take-home midterm (which will be distributed the day you get back from break). Each optional problem is worth a small part of an actual homework problem.

Three of the optional problems are marked with a star (*). These problems are extra credit problems specifically covering the concepts from the midterm that people struggled the most with. Doing all of the redemption problems will give you a modest bump on your first exam grade.

**Required Problems**

**Problem 1  Reduced Fat Skim Milk**

Consider the following proof that \( L = \{0^n1^n | n \geq 0 \} \) is undecidable.

**Claim 1.** \( A_{TM} \) is mapping reducible to \( L \) (\( A_{TM} \leq L \)).

**Proof.** To show that \( A_{TM} \leq L \), we must create a function \( f \) that maps between inputs of a TM that decides \( A_{TM} \) to the inputs of a TM that decides \( L \). Define \( f \) as:

\[
    f(<M, w>) = \begin{cases} 
        000111 & M \text{ accepts } w \\
        000000 & \text{otherwise}
    \end{cases}
\]

Clearly \( L \) is decidable. What is wrong with this proof?

**Problem 2  Electron++**

Prove that \( L = \{<M> | M \text{ is a TM and } L(M) = \Sigma^* \} \) is undecidable.

**Optional Problems**

**Problem 3  New Directions**

(Sisper 5.14 and 5.15) Consider two problems:
1. Determine if a TM $M$ on input $w$ ever attempts to move its head left when its head is on the left-most tape cell.

2. Determine if a TM $M$ on input $w$ ever attempts to move its head left during computation on $w$.

Formulate these two problems as languages $L_1$ and $L_2$. It turns out that $L_1$ is undecidable and $L_2$ is decidable; prove these facts. With what you learned from these proofs, come up with a language $L_3$ that is also a decidable problem that analyzes how a TM $M$ computes on input $w$.

**Problem 4**

Prove that the following languages are undecidable by reduction from a known, undecidable language:

(a) $L = \{ <M> \mid M$ is a TM and the $L(M)$ is finite $\}$.

(b) $L = \{ <M,w> \mid M$ is a TM and during computation on $w$, the substring 010 appears on the tape $\}$.

**Problem 5** Real-world Analysis

Consider the problem of determining if an arbitrary if-expression in Scheme—(if b e1 e2)—produces only one type of value (e.g., only integers or strings). Use a reduction technique from a known undecidable problem to show that this problem is undecidable.

**Problem 6** Reversal

Consider the following proof that $L = \{ <D,w> \mid D$ is a DFA and $D$ accepts $w$ $\}$ is undecidable.

Claim 2. $L \leq A_{TM}$.

To show that $L \leq A_{TM}$, we must create a function $f$ that maps between inputs of a TM that decides $L$ to the inputs of a TM that decides $A_{TM}$. Define $f$ as:

$$f(D, w) = (M, w) \quad \text{where}$$

$$M = \text{"On input $k$: Simulate $D$ on $k$, and do what $D$ does."}$$

We know that simulating a DFA in a Turing Machine is decidable. Therefore, we know that $M$ is a decidable machine. However, $A_{TM}$ is undecidable, so $M$ cannot exist and $L$ must be undecidable as well.

**Problem 7** Decisions, Decisions Redux*

Call a language $A$ *prefix-free* if for all $w \in A$, there does not exist a $w' \in A$ such that $w \neq w'$ and $w'$ is a prefix of $w$. Give a procedure to determine if the language of a DFA $D$ is prefix free.
Problem 8  Closure (Whoops!)*

It turns out that the definition of language homomorphism I gave in the last homework was incorrect! Call this previous definition of homomorphism—transforming a language $L$ via a function $f : \Sigma_1 \to \Sigma_2$—a simple homomorphism. A homomorphism over strings $f$ transforms single characters from the target alphabet to strings in the source alphabet. In other words, $f : \Sigma_1 \to \Sigma_2^*$ (note the difference with this definition of $f$ and the previous definitions).

Prove that the class of regular languages is closed under homomorphism over strings.

Problem 9  Pump You Up Again*

Prove that the language $L = \{0^{2^n} \mid n \geq 0\}$ is not regular.

Problem 10  The Usual Business

Please indicate how much time you spent on this homework. Thanks!